

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-120357

(43)Date of publication of application : 06.05.1997

(51)Int.Cl.

G06F 9/06
G06F 9/44

(21)Application number : 07-278632

(71)Applicant : HITACHI LTD
HITACHI VLSI ENG CORP

(22)Date of filing : 26.10.1995

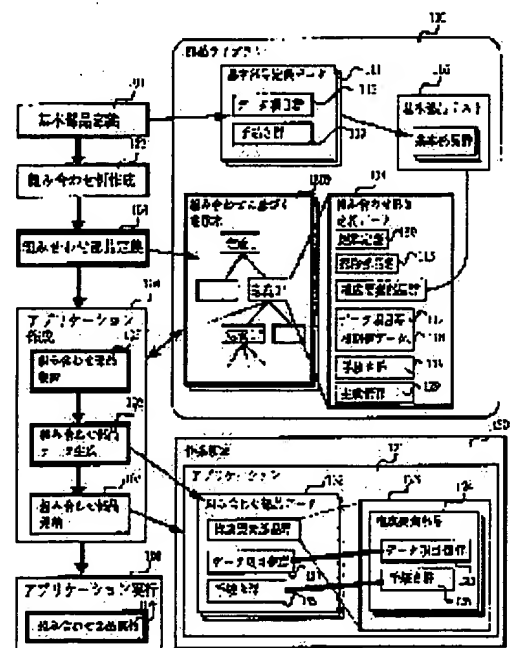
(72)Inventor : KUBO SHOICHI
YUURA KATSUHIKO

(54) OBJECT-ORIENTED APPLICATION PRODUCTION METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To improve the application production efficiency by applying such a procedure that can attain the easy reuse of the combined parts groups as individual parts when the object-oriented instances are used as parts and an application consists of a combination of parts.

SOLUTION: A design system consists of a memory, a CPU and a terminal. The memory includes an object-oriented language processing system, a graphic editor and a parts library 110. The editor 110 has the functions for the combination parts definition, the combination parts retrieval, etc. Then a major parts consisting of a combination of basic parts used for the corresponding application is defined in an object-oriented class of a computer, and the slot and the method of the major parts are related to those of the components parts. Then the major parts is defined, retrieved, generated and carried out based on those slot and method and the initial data on the slot.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-120357

(43) 公開日 平成9年(1997)5月6日

(51) Int. Cl. ⁶	識別記号	庁内整理番号	FI	技術表示箇所	
G 0 6 F 9/06	5 3 0		G 0 6 F 9/06	5 3 0 W	
9/44	5 3 0		9/44	5 3 0 P	

審査請求 未請求 請求項の数 7 O L (全 18 頁)

(21) 出願番号 特願平7-278632

(22) 出願日 平成7年(1995)10月26日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(71) 出願人 000233468

日立超エル・エス・アイ・エンジニアリング株式会社

東京都小平市上水本町5丁目20番1号

(72) 発明者 久保 昭一

東京都小平市上水本町5丁目20番地1号

日立超エル・エス・アイ・エンジニアリング株式会社内

(74) 代理人 弁理士 小川 勝男

最終頁に続く

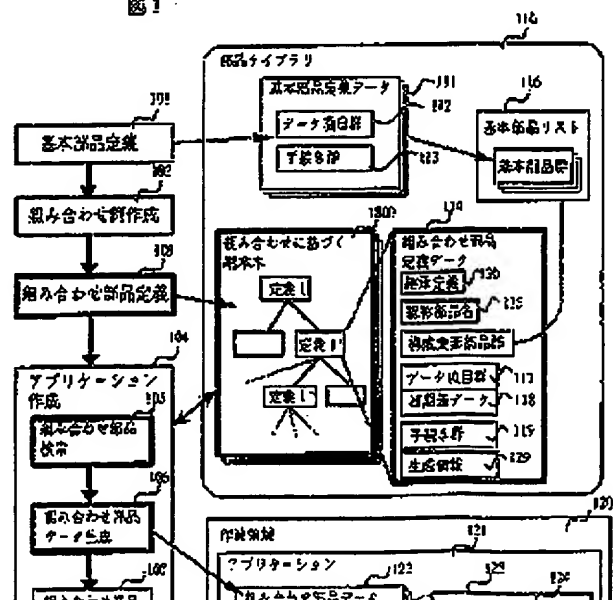
(54) 【発明の名称】 オブジェクト指向アプリケーション構築方法

(57) 【要約】

【課題】 オブジェクト指向のインスタンスを部品として、部品の組み合わせによりアプリケーションを構築する際に、組み合わせた部品群を部品として容易に再利用できるように手順を用いることにより、アプリケーション作成効率の向上を図る。

【解決手段】 計算機上で、対象となるアプリケーションで用いる基本部品を組み合わせた大部品を、オブジェクト指向のクラスで定義し、大部品のスロットとメソッドとを構成要素部品のスロットとメソッドに関係付け、それらスロット・メソッドと、スロットの初期値データをもとに、大部品を定義・検索・生成・実行するアプリケーション構築方法である。

図1



【特許請求の範囲】

【請求項1】1つまたは複数のデータ項目と値を持つデータと前記データを操作する手続き群を有する基本部品をあらかじめ用意し、1つまたは複数の前記基本部品のリストをデータ項目の1つとして持つ組み合わせ部品を作成するシステムにおいて、

オペレータが前記組み合わせに用いられた基本部品群のデータ項目群のうちいくつかを指定し、計算機が当該指定された各データ項目を前記組み合わせ部品の他のデータ項目として定義する組み合わせ部品データ項目定義データ群を生成し、オペレータが前記組み合わせに用いられた基本部品群の手続き群のうちいくつかを指定し、計算機が当該指定された各手続きを呼び出す手続きを前記組み合わせ部品の手続きとして組み合わせ部品手続き群を生成し、

オペレータが部品名を指定し、当該指定部品名から、前記組み合わせ部品と同様の機能を持つ部品群を総称する総称部品名を計算機が生成し、

当該組み合わせ部品に関する前記総称部品名データと、前記組み合わせ部品データ項目定義データ群と、前記組み合わせ部品手続き群の定義データと、継承定義データ、よりなる組み合わせ部品定義データを計算機が生成し記憶する組み合わせ部品定義ステップと、

オペレータが総称部品名と必要とするデータ項目および手続きを指定し、計算機が前記オペレータ指定を満たす組み合わせ部品の定義データを検索する組み合わせ部品検索ステップと、

オペレータが前記検索した組み合わせ部品のデータ項目群に対応する値群を与えて、計算機が前記記憶された組み合わせ部品定義データを参照して前記組み合わせ部品データを生成するとともに前記組み合わせ部品に含まれる基本部品群のデータを生成する組み合わせ部品データ生成ステップと、

オペレータが前記生成組み合わせ部品データのデータ項目群と既に生成してある組み合わせ部品データ群の項目の間でポインタの設定を行う組み合わせ部品連結ステップを有することを特徴とするオブジェクト指向アプリケーション構築方法。

【請求項2】1つまたは複数のデータ項目と値を持つデータと前記データを操作する手続き群を有する基本部品をあらかじめ用意し、1つまたは複数の前記基本部品のリストをデータ項目の1つとして持つ組み合わせ部品を作成するシステムにおいて、

オペレータが前記組み合わせに用いられた基本部品群のデータ項目群のうちいくつかを指定し、計算機が当該指

記組み合わせ部品の手続きとして組み合わせ部品手続き群を生成し、

オペレータが部品名を指定し、当該指定部品名から、前記組み合わせ部品と同様の機能を持つ部品群を総称する総称部品名を計算機が生成し、

当該組み合わせ部品に関する前記総称部品名データと、前記組み合わせ部品データ項目定義データ群と、前記組み合わせ部品手続き群の定義データと、継承定義データ、よりなる組み合わせ部品定義データを計算機が生成し記憶したのち、前記組み合わせ部品データ項目群のうちいくつかのデータ項目の初期値をオペレータが指定し、計算機が初期値定義データ群を生成して、前記総称部品名データ、前記組み合わせ部品データ項目定義データ群、前記組み合わせ部品手続き群および前記組み合わせ部品を継承することを意味する継承定義データを有する別の組み合わせ部品定義データを生成し、記憶する組み合わせ部品定義ステップと、

オペレータが総称部品名と必要とするデータ項目と手続き及び必要に適合するデータ項目の初期値を指定し、計算機が前記オペレータ指定を満たす組み合わせ部品の定義データを検索する組み合わせ部品検索ステップと、オペレータが前記検索した組み合わせ部品のデータ項目群に対応する値群を与えて、計算機が前記記憶された組み合わせ部品定義データを参照して前記組み合わせ部品データを生成するとともに前記組み合わせ部品に含まれる基本部品群のデータを生成する組み合わせ部品データ生成ステップと、

オペレータが前記生成組み合わせ部品データのデータ項目群と既に生成してある組み合わせ部品データ群の項目の間でポインタの設定を行う組み合わせ部品連結ステップを有することを特徴とするオブジェクト指向アプリケーション構築方法。

【請求項3】1つまたは複数のデータ項目と値を持つデータと前記データを操作する手続き群を有する基本部品をあらかじめ用意し、1つまたは複数の前記基本部品のリストをデータ項目の1つとして持つ組み合わせ部品を作成するシステムにおいて、

オペレータが前記組み合わせに用いられた基本部品群のデータ項目群のうちいくつかを指定し、計算機が当該指定された各データ項目を前記組み合わせ部品の他のデータ項目として定義する組み合わせ部品データ項目定義データ群を生成し、オペレータが前記組み合わせに用いられた基本部品群の手続き群のうちいくつかを指定し、計算機が当該指定された各手続きを呼び出す手続きを前記組み合わせ部品の手続きとして組み合わせ部品手続き群を生成し、

前記組み合わせ部品データ項目定義データ群と、前記組み合わせ部品手続き群の定義データと、継承定義データ、よりなる組み合わせ部品定義データを計算機が生成し記憶する組み合わせ部品定義ステップと、

オペレータが総称部品名と必要とするデータ項目および手続きを指定し、計算機が前記オペレータ指定を満たす組み合わせ部品の定義データを検索する組み合わせ部品検索ステップと、

オペレータが前記検索した組み合わせ部品のデータ項目群に対応する値群を与えて、計算機が前記記憶された組み合わせ部品定義データを参照して前記組み合わせ部品データを生成するとともに前記組み合わせ部品に含まれる基本部品群のデータを生成し、オペレータが前記生成されたデータのデータ項目群のうちいくつかを指定し、計算機が前記指定データ項目の値を当該データ項目の初期値とすることを示す定義データを生成し、オペレータが前記組み合わせ部品データ項目群のうちいくつかのデータ項目の初期値を指定し、計算機が初期値定義データ群を生成して、前記総称部品名データ、前記組み合わせ部品データ項目定義データ群、前記組み合わせ部品手続き群および前記組み合わせ部品を継承することを意味する継承定義データを有する別の組み合わせ部品定義データを生成し、記憶する組み合わせ部品データ生成ステップと、

オペレータが前記生成組み合わせ部品データのデータ項目群と既に生成してある組み合わせ部品データ群の項目の間でポインタの設定を行う組み合わせ部品連結ステップを有することを特徴とするオブジェクト指向アプリケーション構築方法。

【請求項4】1つまたは複数のデータ項目と値を持つデータと前記データを操作する手続き群を有する基本部品をあらかじめ用意し、1つまたは複数の前記基本部品のリストをデータ項目の1つとして持つ組み合わせ部品を作成するシステムにおいて、

オペレータが前記組み合わせに用いられた基本部品群のデータ項目群のうちいくつかを指定し、計算機が当該指定された各データ項目を前記組み合わせ部品の他のデータ項目として定義する組み合わせ部品データ項目定義データ群を生成し、オペレータが前記組み合わせに用いられた基本部品群の手続き群のうちいくつかを指定し、計算機が当該指定された各手続きを呼び出す手続きを前記組み合わせ部品の手続きとして組み合わせ部品手続き群を生成し、

オペレータが部品名を指定し、当該指定部品名から、前記組み合わせ部品と同様の機能を持つ部品群を総称する

し記憶する組み合わせ部品定義ステップと、

オペレータが総称部品名と必要とするデータ項目および手続きを指定し、計算機が前記オペレータ指定を満たす組み合わせ部品の定義データを検索する組み合わせ部品検索ステップと、

オペレータが前記検索した組み合わせ部品のデータ項目群に対応する値群を与えて、計算機が前記記憶された組み合わせ部品定義データを参照して前記組み合わせ部品データを生成するとともに前記組み合わせ部品に含まれる基本部品群のデータを生成する組み合わせ部品データ生成ステップと、

オペレータが前記生成組み合わせ部品データのデータ項目群と既に生成してある組み合わせ部品データ群の項目の間でポインタの設定を行う組み合わせ部品連結ステップと、

前記組み合わせ部品定義ステップで定義された組み合わせ部品をあらかじめ用意されている基本部品に代わる構成要素として、前記組み合わせ部品定義ステップと同様の手順でさらに大規模な組み合わせ部品を定義するステップを有することを特徴とするオブジェクト指向アプリケーション構築方法。

【請求項5】請求項1、2、3もしくは4のアプリケーション構築方法によってアプリケーションを構築し、さらに組み合わせ部品および基本部品もしくは構成要素である別の組み合わせ部品の手続きを実行する組み合わせ部品実行ステップにおいて、

組み合わせ部品の手続き実行に伴って組み合わせ部品のデータ項目の値が更新された場合に、

当該データ項目に対応する基本部品もしくは構成要素である別の組み合わせ部品のデータ項目の値を同時に更新し、

構成部品の手続き実行に伴って、基本部品もしくは構成要素である別の組み合わせ部品のデータ項目の値が更新された場合に、

当該データ項目に対応する組み合わせ部品のデータ項目の値を同時に更新することを特徴とするアプリケーション構築方法。

【請求項6】請求項1、2、3もしくは4のアプリケーション構築方法によってアプリケーションを構築し、さらに組み合わせ部品および構成部品の手続きを実行する前記組み合わせ部品実行ステップにおいて、

前記組み合わせ部品定義ステップと同様の手順で、当該組み合わせ部品に含まれる基本部品もしくは構成要素である別の組み合わせ部品がデータ項目もしくはデータ項目値を変えて再定義された場合に、

行方法。

【請求項7】請求項1、2、3もしくは4のアプリケーション構築方法によってアプリケーションを構築し、さらに組み合わせ部品および構成部品の手続きを実行する組み合わせ部品実行ステップを有し、前記組み合わせ部品実行ステップにおいて、

組み合わせ部品連結結果定義ステップの後に、前記組み合わせ部品定義ステップと同様の手順で、当該組み合わせ部品に含まれる基本部品もしくは構成要素である別の組み合わせ部品がデータ項目もしくはデータ項目値を変えて再定義された場合に、

計算機が組み合わせ部品の新しい定義に従って、当該組み合わせ部品連結結果の定義データおよび当該組み合わせ部品連結結果定義データをもとに生成された当該組み合わせ部品連結結果データを更新するステップを有するアプリケーション構築方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】オブジェクト指向のインスタンスを部品として、部品の組み合わせによりアプリケーションを構築する設計システム。

【0002】

【従来の技術】オブジェクト指向言語は、「もの」を部品としてソフトウェアを構成する方法論を提供するものであり、設計問題への適合性が高い。オブジェクト指向言語におけるプログラムの基本要素は、クラスにより定義する。クラス定義では、そのクラスの変数であるスロット群およびそのクラスに関する手続きであるメソッド群を定義する。このクラス定義に従う具体的なプログラム要素をインスタンスという。

【0003】この、オブジェクト指向の考えに基づく設計システムでは、設計対象で用いられる部品の定義を、クラスとしてあらかじめ定義することで、部品ライブラリを構成する。オブジェクト指向設計システムを利用するオブジェクト指向応用プログラム作成ユーザ（以下ではオブジェクト指向応用プログラムをアプリケーションと呼び、アプリケーション作成ユーザをユーザと略す）は、この部品ライブラリ中のクラスから、部品データであるインスタンスを生成し、部品としてアプリケーションに組み込み、アプリケーションを構築する。

【0004】実際のオブジェクト指向設計システムでは、アプリケーションの規模の増大に対応するため、複数の部品を組み合わせた「組み合わせ部品」を提供することで、生産効率の向上を図っている。

【0005】従来のオブジェクト設計システムにおける

義しておき、ここから部品定義を検索し（201）、部品データ209を生成して（202）、部品どうしを連結して、組み合わせ例208を作成する。作成した組み合わせ例は、他のアプリケーション開発で再利用できるように、二次記憶装置などに、まとめて保管される。アプリケーション作成時には、組み合わせ例のデータ群207から、組み合わせ例208を検索して、構築対象のアプリケーションに合うように編集を行い、アプリケーション121に組み込み、生成した組み合わせ例の部品データどうしを連結して（212）、アプリケーションを作成していく（203）。

【0006】そしてデバッグのための試用あるいは実際の利用のためのアプリケーション実行206を行う、というようにして、アプリケーションの構築を行っていた。

【0007】

【発明が解決しようとする課題】従来技術では、組み合わせ部品の再利用、保守、安全性、適用性に関して配慮されておらず、以下の様な問題点があった。

【0008】〔問題点1〕組み合わせ例の編集ミスの負荷大

部品組み合わせ例を再利用する方法では、組み合わせ例からアプリケーションを作成していく過程で、どの部品のデータ項目を修正し、どこにどういった部品を追加していくかを定める手順がユーザにとって分かりにくく、かつ修正追加項目が多い。このため修正追加の編集ミスが多発する。

【0009】〔問題点2〕組み合わせ例の実行時の動作不良

アプリケーション実行時に、外部（すなわち他の組み合わせ例部品）より直接、要素部品のデータ項目や手続きをアクセスするためには、要素部品データ自体をいったん一時変数に取り込み、アクセスしなければならない。このため、要素部品のデータ項目や手続きのうち、組み合わせ例の外部からアクセスしては困るようなものも、誤ってアクセスしてしまう場合がある。

【0010】〔問題点3〕共通管理部分の一貫性の保持が困難

以前に作成した組み合わせ例を修正して新たな組み合わせ例を作る場合、組み合わせ例の間の共通部分が数多く生まれる。このような共通部分は、各組み合わせ例の間で共通の意味で使われる場合が多く、そのようなとき、修正が必要な場合はすべて同じく修正されるべきであるが、従来の方法では、各組み合わせ例を一つ一つ修正しなければならないので、手間がかかり、修正ミスがあった場合には、動作不良の原因になる。

10

20

30

40

手間がかかるうえ、必要な機能をもつ部品を正しく換装できるとは限らなくなる。

【0012】本発明の目的は、上記問題点を解決する組み合わせ部品の作成・利用手順とデータ構造を与えることで、オブジェクト指向アプリケーションの作成効率を向上させる方法を提供することにある。

【0013】

【課題を解決するための手段】

(1) 組み合わせ部品定義

構成要素部品のデータ項目群の一部を組み合わせ部品のデータ項目群と対応付けるためのデータと、構成要素部品の手続き群の一部を組み合わせ部品の手続き群と対応付けるデータと、前記組み合わせ部品のデータ項目群の初期値データからなる組み合わせ部品を定義し、定義した組み合わせ部品と同種の機能をもつ組み合わせ部品群を総称する総称部品名を、当該組み合わせ部品群に定義する。

【0014】(2) 継承による系統化

組合せ部品をオブジェクト指向のクラスで表現し、同じ総称部品名をもつ部品間でデータ項目・初期値・手続 20
きの構成に基づき、継承木を自動生成する。

【0015】(3) 検索

オペレータが総称部品名を指定することで、計算機が総称部品名に対応する継承木を検索し、継承木をもとに、データ項目、初期値、手続きの組み合わせの検索・指定を行うことで、組み合わせ部品の検索を行う。

【0016】(4) 基本部品と組み合わせ部品の一貫性管理

部品の再定義時に、オブジェクト指向言語処理系が行うところの、古い定義に基づく部品を新しい定義に適合するよう更新する処理に対し、当該部品を含む組み合わせ部品の連結データを変更して再定義する処理と、構成要素部品データと組み合わせ部品データの連結を変更する処理を追加する。

【0017】問題点1に対し、手段(1)により、組み合わせ部品のデータ項目群という形で、カスタマイズして意味のあるデータ項目が与えられ、かつ、手段(2)により機能や構成のバリエーションが系統的に与えられるので、必要な機能・構成を持つ組み合わせ部品を検索して、データ項目値を設定するだけで、意味のある組み合わせ例が得られるので、組み合わせ例の編集が不要となり、組み合わせ例の編集ミスの負荷が低くなる。

【0018】問題点2に対し、手段(1)により、組み合わせ部品の内部でしか用いないような、要素部品のデータ項目や手続きは、組み合わせ部品の要素部品のデータ項目や手続きとして定義される必要がある。

正なアクセスによる、組み合わせ例実行時の動作不良がなくなる。

【0019】問題点3に対し、手段(2)と(3)により、共通部分が一つのクラスとして定義され、共通部分を一括変更したいときは、クラスの再定義により部品が自動的に更新される。これにより、一貫性は自動的に保たれる。したがって、一貫性を保つために要する手間と、一貫性が保ちきれなかったために起きる動作不良がなくなる。

【0020】問題点4に対し、手段(2)と(3)により、継承木にしたがって、機能(手続き)やデータ項目の構成であらかじめ分類されているので、検索が容易かつ正確に行える。

【0021】したがって、各手段で与えられた組み合わせ部品の作成・利用手順とデータ構造により、アプリケーション作成効率の向上という目的が達成される。

【0022】

【発明の実施の形態】図1は、本発明の一実施例に係わるアプリケーション構築手順である。本実施例で用いる設計システム(図3)は、メモリ301、CPU304、端末305から構成される。メモリ301には、オブジェクト指向言語処理系303、図形エディタ302、部品ライブラリ110が搭載されており、部品エディタ110には、組み合わせ部品定義306、組み合わせ部品検索307、組み合わせ部品データ生成308、部品連結309、組み合わせ部品実行310の各機能がある。このうち、図形エディタ302、部品連結309の各機能については、「情報処理学会、オブジェクト指向ソフトウェア技術シンポジウム論文集pp.1-11(1991)、ODETTE:オブジェクト指向CLOSをベースとした設計支援構築環境」に述べられている。

【0023】また、オブジェクト指向言語処理系とは、CLOS(Common Lisp Object System)の仕様を満たすような言語処理系であり、クラス定義、インスタンス生成、終結関数制御、スロットアクセス等の機能を有する。CLOSの言語仕様は、D.G. Bobrow, et. al., "Common Lisp Object System Specification", ANSI X3J13 Document 88-002R(1988)に詳しい。

【0024】本実施例においては、アプリケーションの構築に先立ち、組み合わせ部品定義を、オブジェクト指向の「クラス」として、図6のように構成する。以下これを組み合わせクラスと呼び、組み合わせクラスより生成される組み合わせ部品データを組み合わせインスタンスと呼ぶことにする。

【0025】以下、図1のアプリケーション構築手順について、図8から図18を用いて具体的に説明する。

て持つ装置801や、タンク802、ライン803、調整装置804を作成し、部品ライブラリ110中に登録する。これらは計算機によりアイコン形式で、部品リスト画面901に一覧表示される。

【0027】〔手順2〕組み合わせ例作成102

次の2つの作業の繰り返しにより部品の初期の組み合わせ例を部品エディタ画面902上に作成する。

【0028】作業1：オペレータが、部品リスト画面901から、アイコンをドラッグ・ドロップ操作で、部品エディタ画面902上に配置する。

作業2：2つの部品間で、オペレータが連結操作を行うと、部品連結機能309により、部品間のデータ授受の経路が設定される。

【0029】これにより、図9のように、造粒装置、造粒タンク、荷造りラインからなる組み合わせ例が作業領域に生成される。

【0030】〔手順3〕組み合わせ部品定義103

組み合わせ例に対し、オペレータが総称部品名115を指定する。これにより、組み合わせ部品定義データ114が計算機により生成される。具体的には、図9のように、ダイアログ画面903で総称部品名の入力を行う。この操作により、部品ライブラリ中には、図18のように、前記総称部品名と、構成要素部品群の生成データ602として、装置・タンク・ラインの各生成データ1808を持つ組み合わせクラス1801が計算機により生成される。

【0031】次に、構成要素部品のスロット、メソッドのうち、組み合わせ部品のスロット、メソッドとして設定したいものをオペレータが指定する。図10は、造粒タンク1001の最大容置スロット1002を、変数設定ダイアログ画面1004を用いて、造粒ラインクラスの内蔵容置という名前のスロット1003として指定する例を示している。図11は、造粒装置1101の速度設定メソッド1102を、造粒ラインクラスの造粒速度設定メソッド1103として、メソッド設定ダイアログ画面1104を用いて指定する例を示している。

【0032】これにより、計算機は、前記組み合わせクラス1801の子クラスとして、スロット117と、スロットと構成要素部品スロットの接続関係の定義603、およびメソッド119と、メソッドと構成要素部品メソッドの接続関係の定義604が追加された構造のクラス1803を部品ライブラリ110中に生成する。

【0033】また、既に作成済みの組み合わせクラスを編集して、組み合わせのバリエーションを増やすことが出来る。

【0034】12のように、オペレータが調整装置1201

【0035】さらに、図10のように、オペレータが初期値の指定1005をすることで、部品ライブラリ110には、前記クラス1803に初期値データを加えた子クラスの定義データ1805が計算機により生成される。

【0036】以上のようにして、部品ライブラリ110中には、継承木1800が構築される。

【0037】〔手順4〕組み合わせ部品の継承木を用いたアプリケーション作成104

(1) 組み合わせ部品検索105

10 手順3が終わると、図13のように、計算機は、部品リスト画面901上に、部品群の一つとして、総称部品名と、アイコンを表示する。(1301)。

【0038】組み合わせ部品を生成するために、オペレータは、造粒ラインの組み合わせ部品1301を基本部品と同様にドラッグ・ドロップ1302により部品エディタ画面上に配置する(部品エディタは、アプリケーションの作成にも用いられる)。このとき、検索ダイアログ画面1303により変数(すなわちスロット)群の選択1304と、初期値群の選択1305、メソッド群の選択1306をして造粒ラインで総称されるクラス群から所望のクラスを検索し、変数値設定ダイアログ画面1307により変数値入力領域1308に値を入力して、生成する組み合わせインスタンスのスロット値の設定を行う。これにより、図14のような組み合わせインスタンス1401が生成される。ここでは、内蔵容置スロット1402が、タンクの最大容置スロット1403と連結され、生産量スロット1404が、調整装置の生産量スロット1405と連結されるような、組み合わせインスタンスが生成された場合の例を示している。

【0039】図15のように検索ダイアログ画面1303上で、現在選択している組み合わせクラスに対してさらに初期値設定1501をした場合には、初期値データを追加した新たな組み合わせクラス1806が部品ライブラリ追加される。

【0040】(2) 組み合わせ部品データ生成106

計算機は、検索した組み合わせ部品定義データから、組み合わせ部品データ生成106を行なう。組み合わせ部品データ生成では、図5のように、オペレータが組み合わせ部品のデータ項目に対応する値を与えると、計算機は、組み合わせ部品に含まれる構成要素部品群のデータを生成し(501)、生成した組み合わせ部品データ122のデータ項目群127と構成要素部品124のデータ項目群125との連結を行い(502)、さらに組み合わせ部品データ122の手続き群128と構成要素部品の手続き群126の連結を行う(503)。

【0041】入力したデータ項目値のうちいくつかを初

値入力領域1308に値を入力して、初期値指定ボタン1501を押して、初期値を指定操作を行うことにより、新たに初期値設定を持つ組み合わせクラス1807が計算機により生成され、部品ライブラリ110に追加される。

【0043】組み合わせ部品データが生成されるときは、計算機内部の処理を、図6を用いて詳しく説明する。組み合わせインスタンス生成時には、生成時処理608が呼び出され、以下の手順で処理が行われる。

【0044】処理1：構成要素部品群の生成データ602にもとづいて部品群生成609が実行され、基本部品リスト115として格納される。

処理2：当該部品のスロットと構成要素部品スロットの連結情報609にもとづいて構成要素部品のスロットとの連結処理610が実行され、データ項目（スロット）群117のうちいくつかと、構成要素部品のスロットが連結される。

【0045】処理3：当該部品のメソッドと構成要素部品メソッドの連結情報609に基づき、構成要素部品のメソッドとの連結処理611が実行され、手続き（メソッド）群118のうちいくつかと、構成要素部品のメソッドが連結される。

処理4：初期値データ118に基づき、初期値設定612が実行され、データ項目群117のうちいくつかの初期化が行われる。

処理5：親クラスの生成時処理がよびだされる（613）。

【0046】（3）組み合わせ部品連結107

手順2の作業2と同様にして、生成した組み合わせ部品と他の部品との連結を行う。

【0047】〔手順5〕組み合わせ部品を用いたアプリケーションの実行108

以上のようにして作成したアプリケーションの実行108の間に生じる組み合わせ部品の各手続きの実行109において、手続き実行に伴って組み合わせ部品のデータ項目の値が更新された場合には、データ項目に対応する構成部品のデータ項目の値が同様に更新される。また、構成部品の手続き実行に伴って、構成部品のデータ項目の値が更新された場合には、データ項目に対応する組み合わせ部品のデータ項目の値が同様に更新される。

【0048】具体的には、図14のように、作成したアプリケーションの実行時に、造粒ライン1401の内蔵容量スロット1402が更新された場合には、タンク1403の最大容量スロット1403の値1406が同時に更新される。同様に、タンク1403の最大容量スロット1403の値1406に変更があった場合には、同時に造粒ライン1401の内蔵容量ス

れ（701）、各組み合わせクラスについて再定義が行われ（702）、オブジェクト指向言語処理系によって、再定義された部品定義データにしたがって構成要素部品データ群と組み合わせ部品データの変更が行われる。

【0050】以上のようにして作成したアプリケーションを新たな組み合わせクラスとして階層的に定義する場合は、図17のように作成したアプリケーションの画面1701に対してダイアログ画面903に部品名を入力することにより、手順3の部品名称の入力操作の次の操作を繰り返せばよい。

【0051】以上に示した本実施例によるアプリケーション構築手順は、以下のような特徴を有する。

【0052】（1）機能・構成に基づく組み合わせ部品の検索と、データ項目値の設定による組み合わせ例取得手順4のように、造粒ラインの構成要素を直接編集しなくても、造粒ラインを検索・カスタマイズして利用できる。

【0053】（2）他の組み合わせインスタンスからのアクセスが無意味なデータ項目の隠蔽

図8のタンク802の現在値スロットは、他の組み合わせインスタンスから変更しても意味のないパラメータであり、組み合わせ部品定義時に組み合わせ部品のスロットとして指定しなければ、このスロットは他の組み合わせインスタンスのアクセスから隠蔽される。また、タンク802の注入・排出メソッドも、他の組み合わせインスタンスから直接呼び出す必要のないメソッドであるとして、作成の段階で隠蔽できる。

【0054】（3）継承による共通部分の一元管理

図18の組み合わせクラス1805と1804は、共通の親クラス1803を持つ。手順5で示したように、親クラスの定義を変更すれば、子クラス1804、1805の定義も計算機によって一括して変更されるので、ユーザが特に管理しなくても、自動的に一貫性が保たれる。

【0055】（4）継承木による機能（手続き）・データ項目の構成の分類

手順3により、同じ総称部品名をもつ部品に関して、最初に定義した組み合わせから出発して、定義の継承を順次行なうことにより、変数や構成要素初期値の組み合わせに基づいた継承木1800が部品ライブラリ中に構成され、さらに、手順4で示したような画面上での検索により容易に利用できる。

【0056】

【発明の効果】

（1）組み合わせ例の再利用効率の向上

必要な機能・構成を持つ組み合わせ部品を検索して、デ

例) からアクセスされては困るようなものが組み合わせ部品の内部に隠蔽されるので、不正なアクセスによる動作不良がなくなる。

【0058】(3) 一元化による一貫性保持

継承により共通部分を一元管理できるので、一貫性を保つために要する手間と、一貫性が保ちきれなかったために起きる動作不良がなくなる。

【0059】(4) 検索効率向上

継承木にしたがって、機能(手続き)やデータ項目の構成であらかじめ分類されているので、検索が容易かつ正

【図面の簡単な説明】

【図1】本発明のアプリケーション構築手順である。

【図2】従来技術のアプリケーション構築手順である。

【図3】本発明の一実施例の構成である。

【図4】組み合わせ部品定義の手順である。

【図5】組み合わせ部品データ生成の手順である。

【図6】組み合わせクラスの構成である。

*

*【図7】再定義時の起動メソッドの構成である。

【図8】基本部品の定義例である。

【図9】組み合わせ例の作成操作である。

【図10】変数初期値の指定操作である。

【図11】メソッドの指定操作である。

【図12】組み合わせ例の編集操作である。

【図13】組み合わせインスタンスの生成操作である。

【図14】組み合わせインスタンスの構造である。

【図15】検索時の初期値設定操作である。

【図16】変数値設定時の初期値設定操作である。

【図17】階層定義操作である。

【図18】組み合わせに基づく継承木である。

【符号の説明】

103 - 組み合わせ部品定義

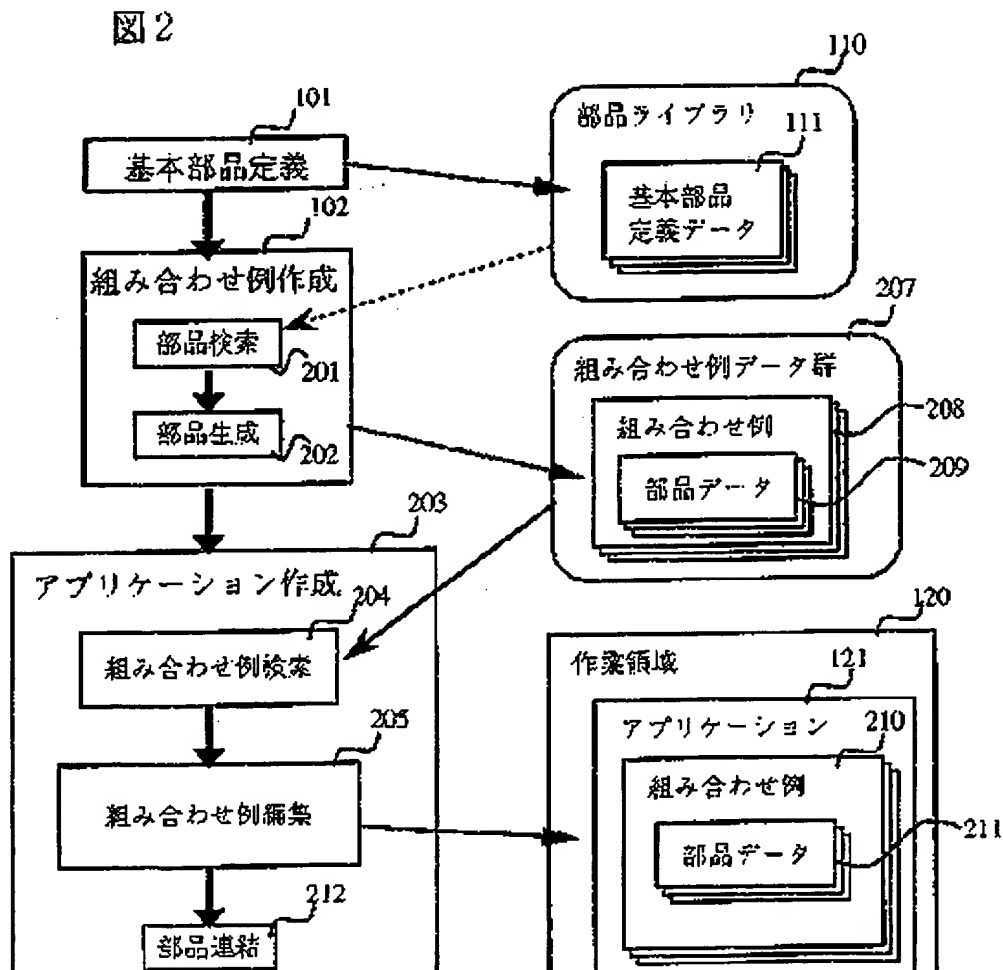
105 - 組み合わせ部品検索

106 - 組み合わせ部品データ生成

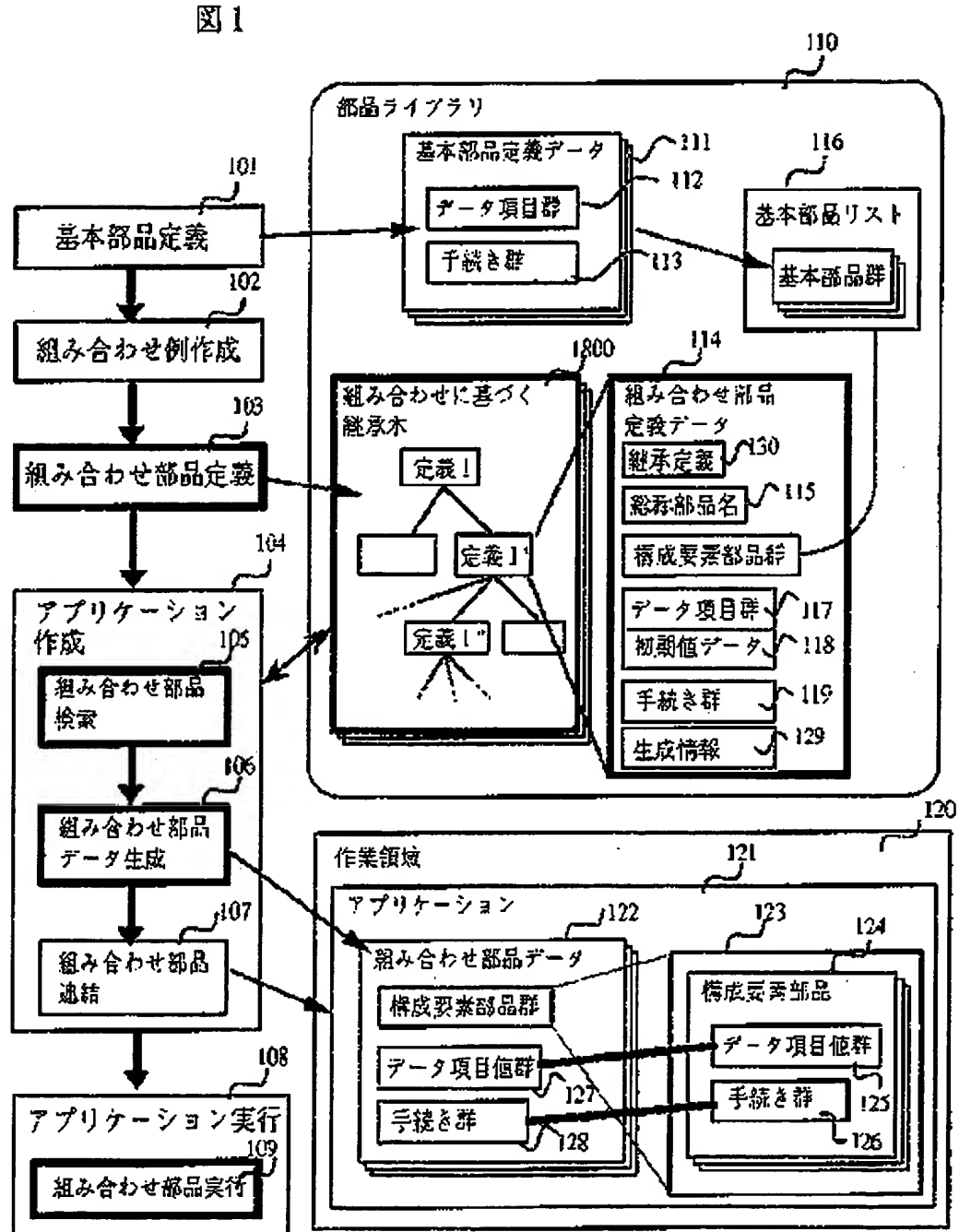
109 - 組み合わせ部品実行

* 110 - 部品ライブラリ

【図2】



【図1】



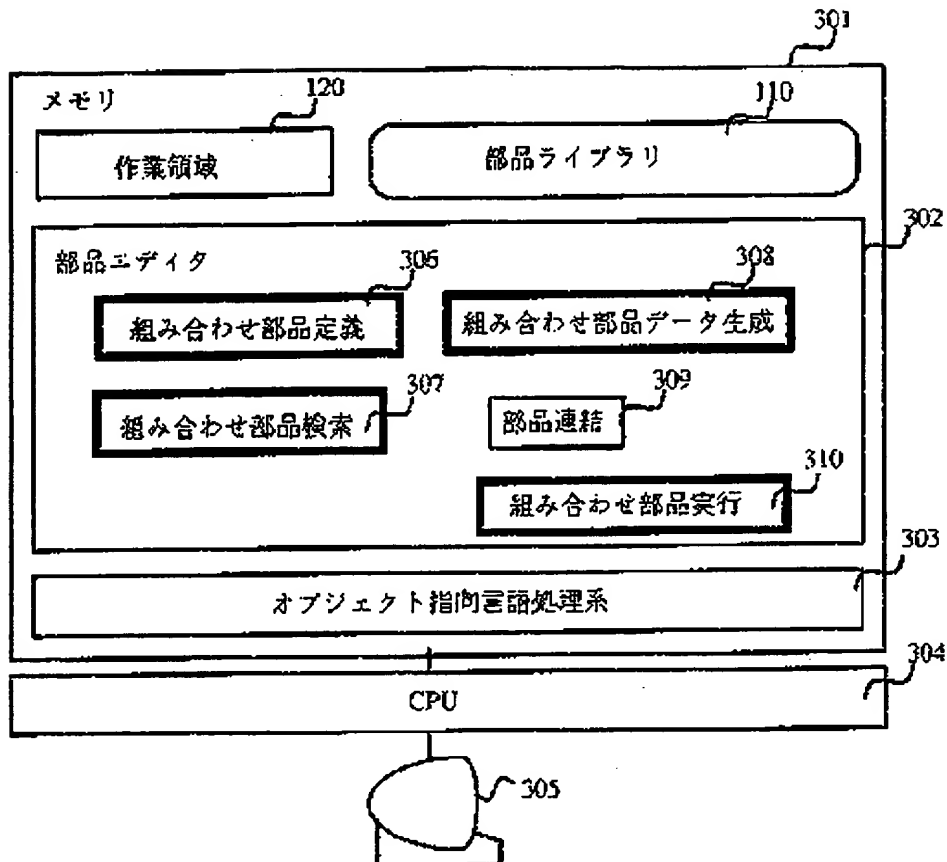
【図7】

図 7

```
(defmethod update-instance-for-redefined-class :after ((instance 部品) ....)
```

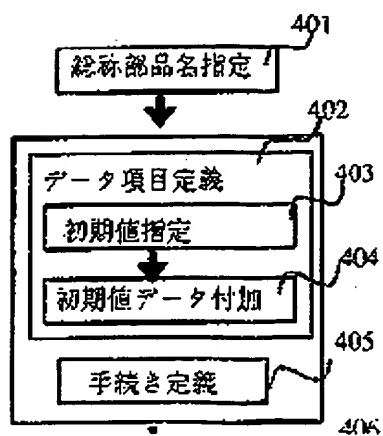
【図3】

図 3



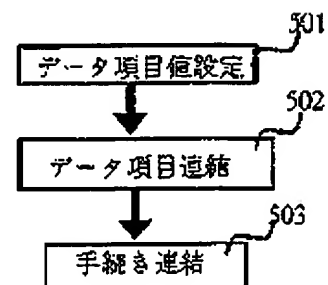
【図4】

図 4



【図5】

図 5



106

【図6】

図 6

130	継承するクラス	(造粒ライン-initial)
115	総称部品名	(generic-name '造粒ライン)
116	構成要素部品群	(components :uniform nil)
117	データ項目群	最大容量 ...
118	初期値データ	(default-values :uniform ((最大容量 100) ...))
119	手続き群	(defmethod 速度設定 ...)
129	生成情報	
603	当該部品のスロットと 構成要素部品スロット の連結情報	(slots :uniform ((最大容量 (2 capacity))))
604	当該部品のメソッドと 構成要素部品メソッド の連結情報	(methods :uniform ((速度設定 造粒速度設定) ...))
602	構成要素部品群の 生成データ	(component-prototypes :allocation class :uniform ((造粒装置 ...) (造粒タンク ...) ...))
608	生成時処理	(defmethod initialize-instance :around ((object 造粒ライン) &rest initargs) 部品群生成 構成要素部品のスロットとの連結処理 構成要素部品のメソッドとの連結処理 初期値設定 親クラスの生成時処理をcall)

609

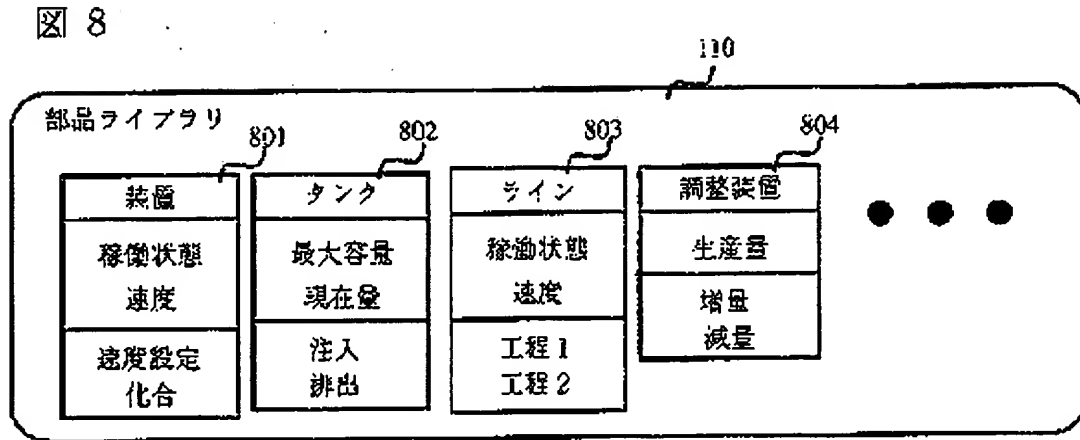
610

611

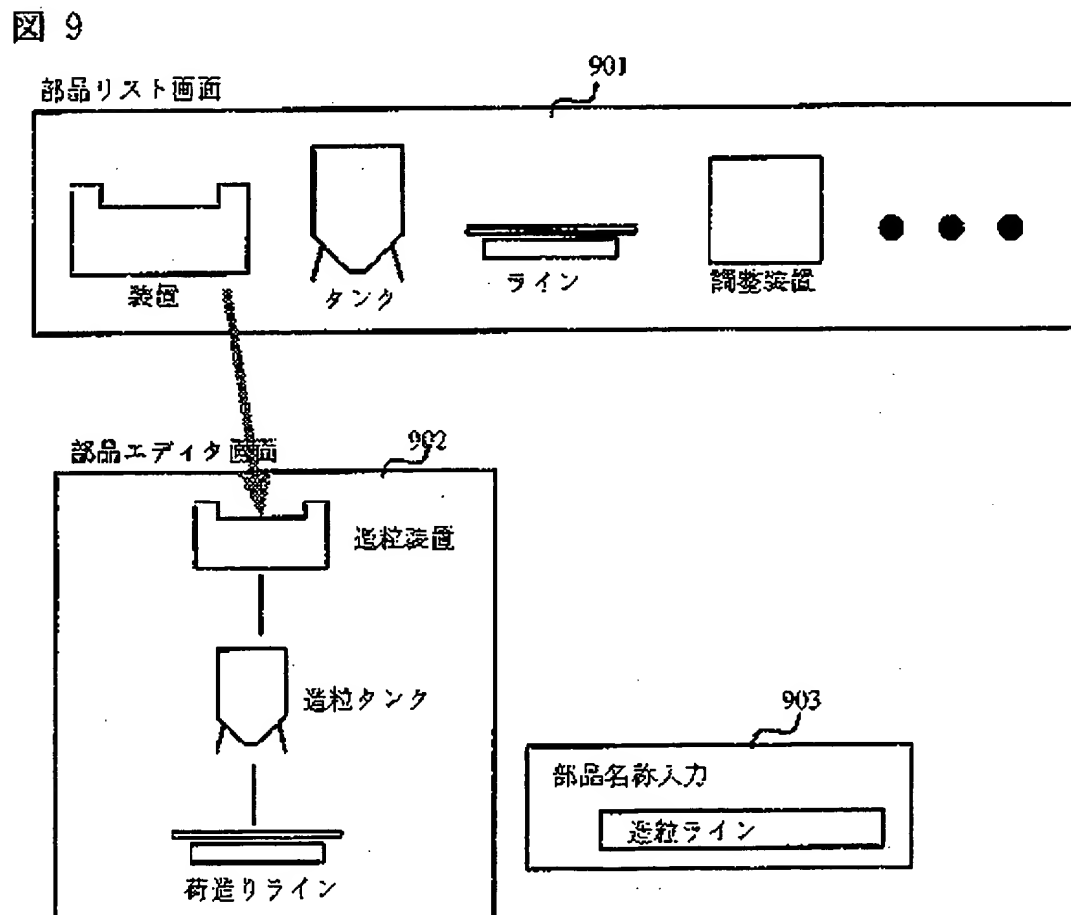
612

613

【図8】

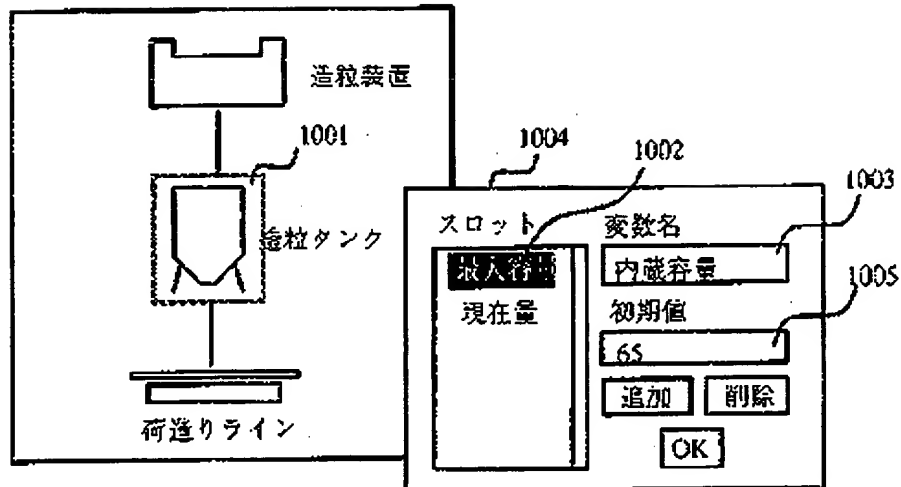


【図9】



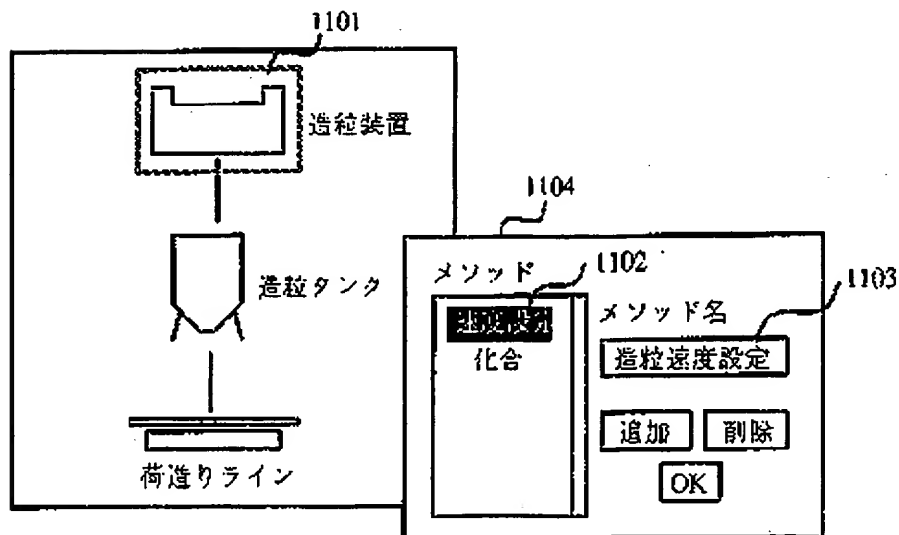
【図10】

図 10



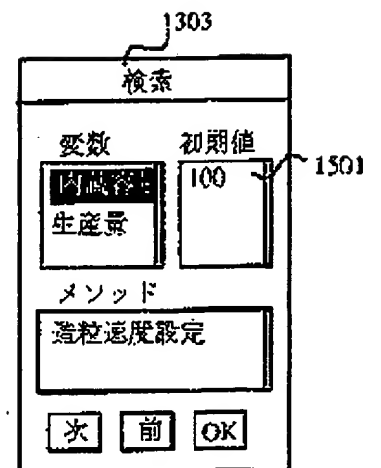
【図11】

図 11

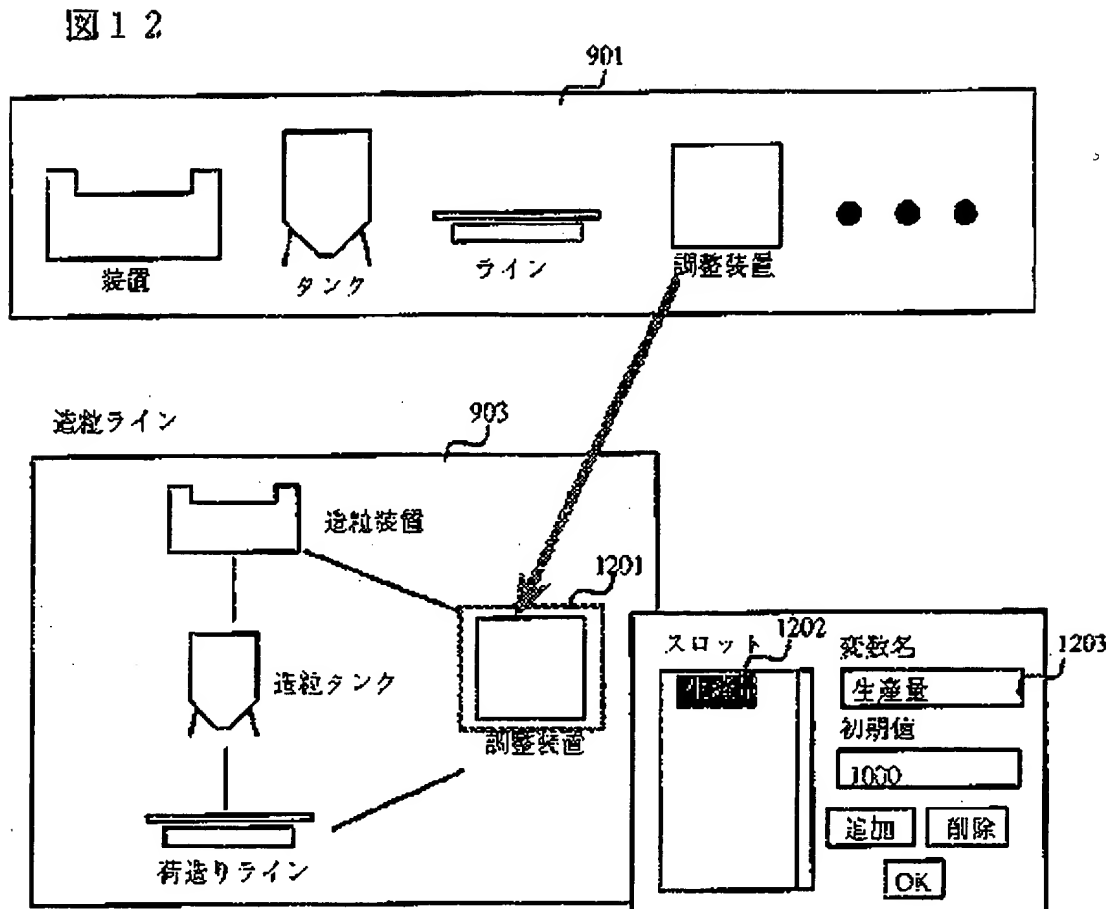


【図15】

図 15

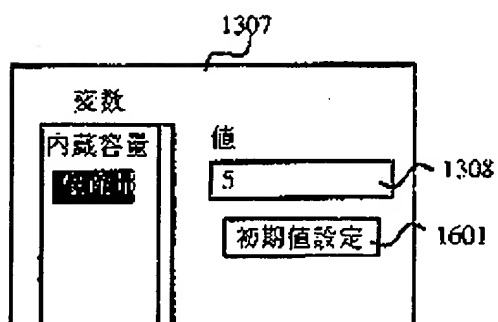


【図12】



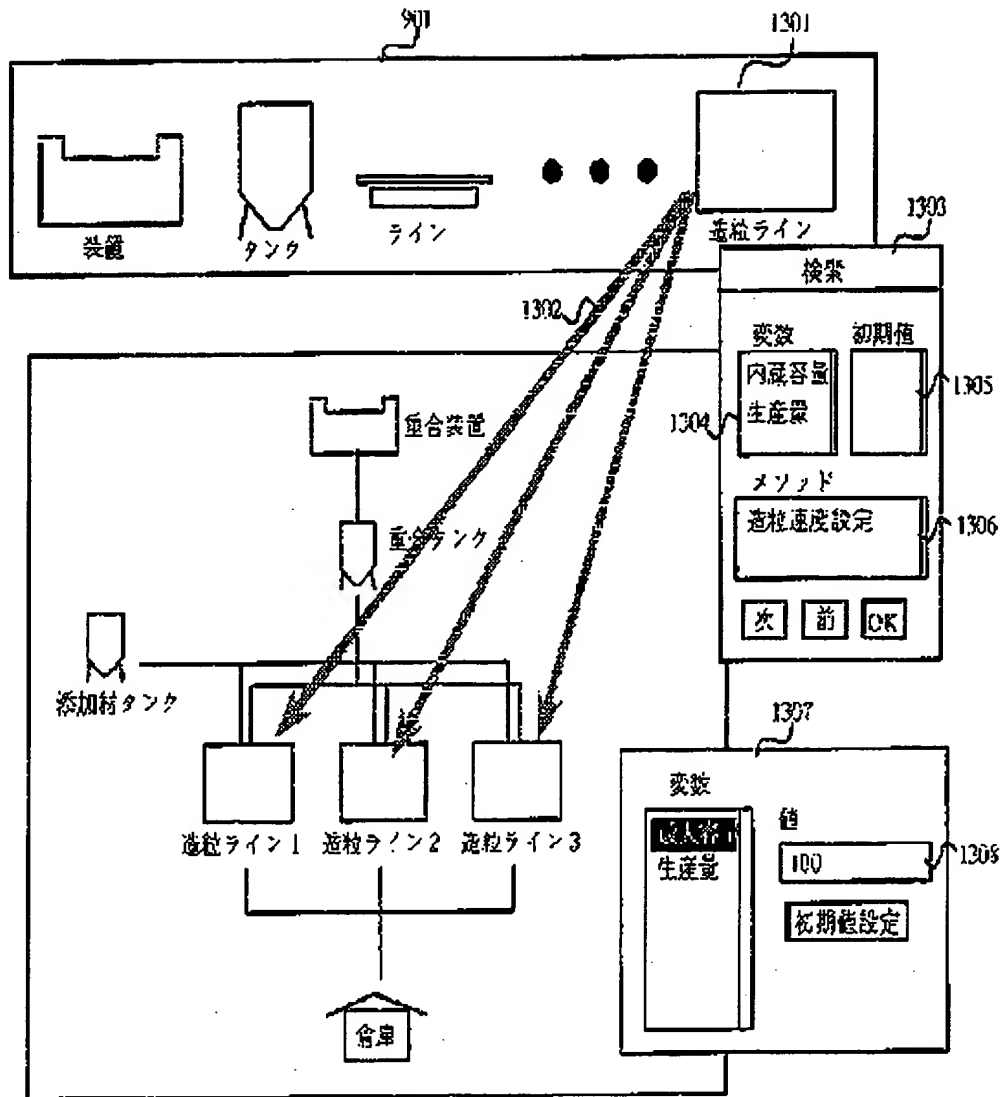
【図16】

図16

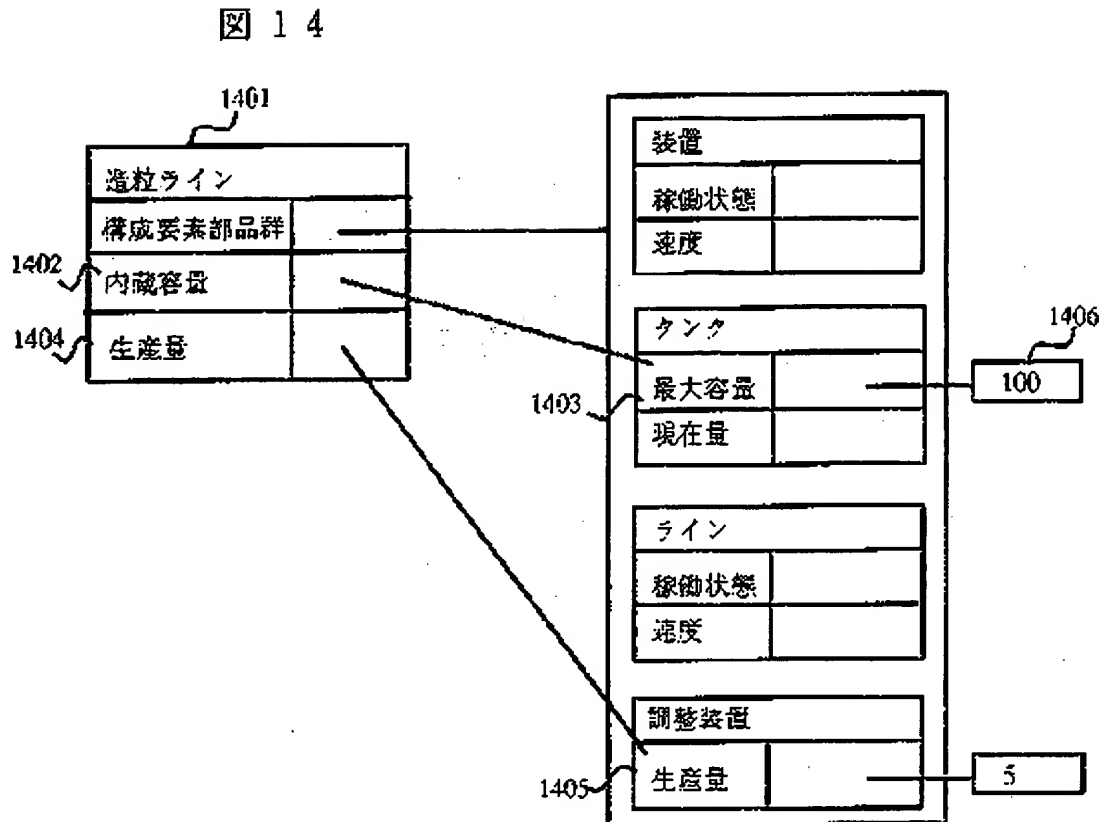


【図13】

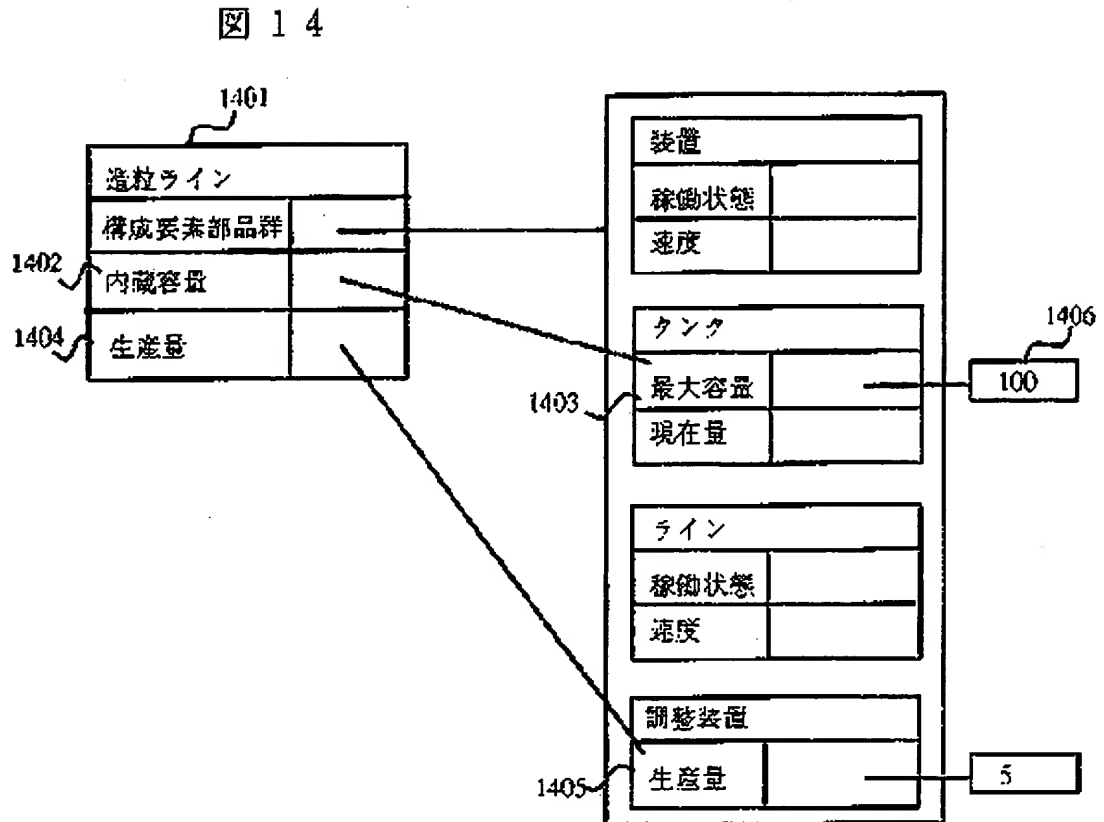
図 13



【図14】

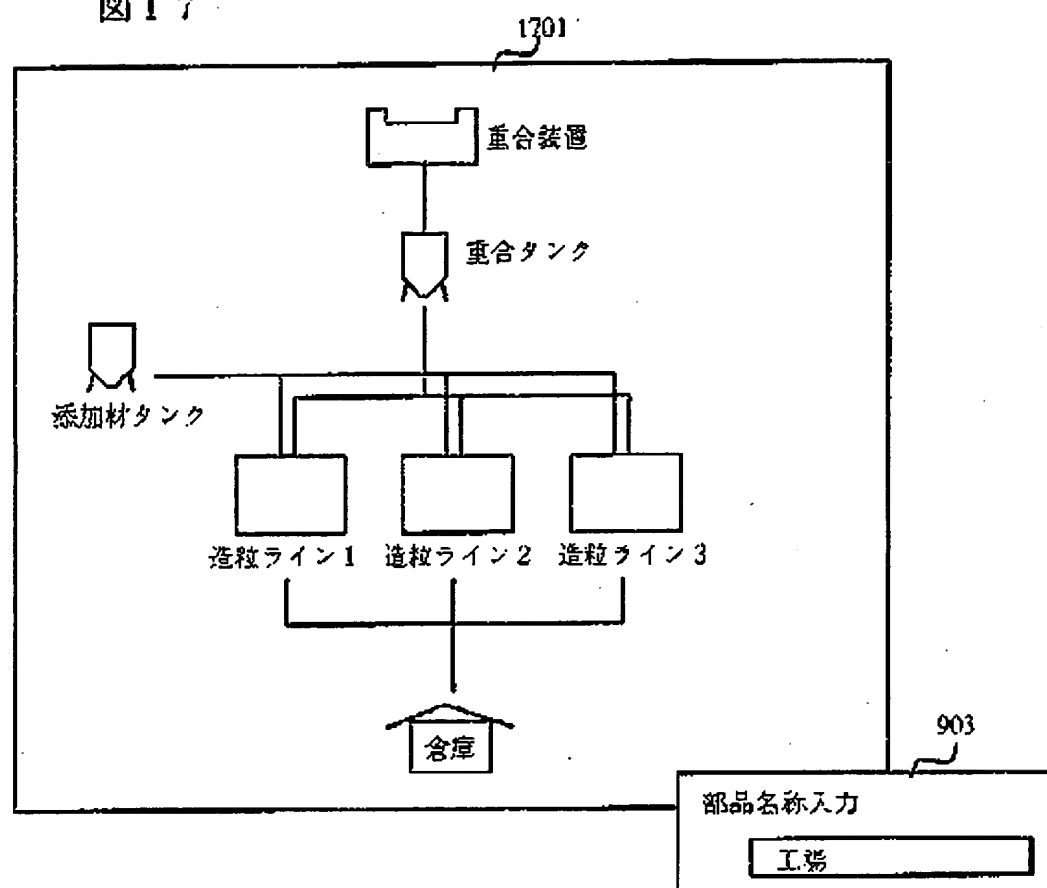


【図14】



【図17】

図17



【図18】

